8704/8705 (032)
Central Tex

The Paper Peripheral

Central Texas 99/4A Users Group
General Membership Meeting
February 12, 1987

The February meeting of the Central Texas 99/4A Users Group was held on Thursday, February 12, 1986 at the Commissioner's Court Room of the Travis County Courthouse Annex building. The meeting was called to order at 7:45pm by the President, Joe Pizzi.

The minutes of the January meeting were read by the Secretary, Mike Schultz.

The Treasurer, Paul Dunn, made a treasurer's reoort. He reported that the account was too small to qualify for free checking at banks. He said that he would check with Credit Unions next month.

Joe Pizzi introduced the group's officers.

Hank Kennedy reported on his efforts to contact the group's members.

Hank also said that he would try and get some presentations for the group. One idea is a hardware upgrade session.

Finally Hank announced that he had several articles that people could have.

The times of the SIGs were announced.

The meeting adjourned at 8:01pm.

Mike Schultz
Secretary

Central Texas 99/4A Users Group
General Membership Meeting
March 12, 1987

The March meeting of the Central Texas 99/4A Users Group was held on Thursday, March 12, 1986 at the Commissioner's Court Room of the Travis County Courthouse Annex building. The meeting was called to order at 7:53pm by the President, Joe Pizzi.

The minutes of the February meeting were read by the Secretary, Mike Schultz.

The Treasurer, Paul Dunn, made a treasurer's report. He also reported that he checked with several banks and credit unions about the account. He found that they wouldn't handle the account as an individual account and handling it as an corporate account would mean a service charge.

Foley Broaddus accounced that he had an experiment that he wanted to conduct and asked the membership for their help. He wanted to find out which disk drives could read flippies without having the second sync hole punched out. He had some flippies without the second hole and asked people to take them home and try them out. He also had a form to fill out and mail back to him.

Jeff Kenner had a presentation about the features of the Triton Bridge box between a 99/4A and an IBM PC. He hoped to have a real one next month to show.

Joe Pizzi announced that Wayne Smith was having problems with a modem and was seeking help. Wayne also had a TI 771 and computer desks for sale.

Mike Schultz talked about the Brown Symposium at Southwestern University. The focus of the symposium was to be computers in everyday life.

John Modgling suggested a new place to hold the meetings. It is where he works and has a place to hold the meetings. The locations is North Lamar. The meeting time would have to be changed to Saturday but that would leave more time to bring and setup equipment and more time afterwards for discussion. The officers stated that they would investigate the location.

John Kennedy announced that he had information about increasing the quality of the printing of a Epison printer by changing the ROMs in them.

John also announced that at the next meeting, there'll be a session to upgrade 99/4A consoles to let the joysticks be used with the Alpha Lock key up.

The meeting adjourned at 8:30pm.

Mike Schultz
Secretary

FROM THE ADVENTURERS SIG: The latest happinings, Lori Milam reporting.

Well nobody showed up for the February meeting, so we didn't do anything in February.

Well March was a much better turn out, we had Keith Johnson, Joe Pizzi, and Kathy Puckett (and son).

We worked on Sorcerer, but to no avail.
We wandered all over the guild, and got stuck in the dream when we went to sleep. At this point we gave up! So if anyone has gotten past this point, LET US KNOW!!
This is a reminder that the adventurers SIG meets at Lori Milam's house the sunday after the main meeting, the scheduled dates are: April 12, and May 17.
If you need directions, call any of the officers.

TRANSMISSION RECEIVED AND DECODED
STAR DATE: 870330.20
SOURCE: USS HAMILTON
ORIGINATOR: KAPTAIN KLUDGE
DESTINATION: CENTRAL TEXAS 99/4A USERS GROUP, AUSTIN, TX, USA, TERRA, SOL

MESSAGE CONTENTS FOLLOW:

I have completed mission on planet UNIX and en route back to Terra.

Plan to have class prepared on C ready for the next Assembly Language SIG.

Currently planning to cover file I/O. Hope that you can be there.

Also plan to resume newsletter articles, unless I'm called to active duty again.

Till then,
The Kaptain of Kludges

END OF TRANSMISSION

Submitting a Group of Articles from other UG'S; MICHAEL JONES

Condensed from LA 99'ers JAN.85/via LEHIGH 99'ER COMPUTER GROUP

This program delets a large number of unwanted lines from a program. The problem may arise when you have a long program wich conains a subroutine you wish to save. If you do not have TI's Programming Aids III, you must delet lines one at a time (quite time consuming as well as boring), list to a printer and retype only the lines you want to save (subject to error), or list to a disk and edit out the unwanted lines with TI Writer then translate back to a program. The attached program will request beginning and ending line numbers of thr routine wich you wish to save and will do the job automaticiil. Its speed of operation depends solely on the distance from highest line in the program to the lowest line to be saved. It checks about ten lines per second. This program can be used in EXTENDED Basic only. It should be stored in MERGE format and then merged with the program ypu wish to dismember.If there is a line above 7 wich you wish to be saved, your program should be resegenced before merging this. RUN the program. When it stops, LIST. You will have only the lines you asked to be saved. You should save the shortened program using MERGE so that the unused lines, which are still in memory are not saved.
How can this program wipe itself out while running? The Operating System refers to the Line Pointer Table only when it is looking for another line of Basic. Regardless of what else is happening, it continues on to the end of the current line unless directed elsewhere. So, the entire routine wich makes the change is in line 6, the rest of the program is just calculations.

SUBROUTINE EXTRACTOR by the author GEORGE F. STEFFEN

```
1 CALL CLEAR :: CALL INIT ::
INPUT "Line numbers of rout
ine to be saved: First, La
st? ":l,m :: G=256 :: CA
LL PEEK(-319252,H,I,J,K)
2 C=INT(M/G):: D=M-C*G :: F=
(J-G)*G+K :: FOR E=(H-G)*G+I
```

```
TO F STEP 4 :: CALL PEEK(E,
A,B):: IF A=C AND B=D THEN 4
3 NEXT E :: PRINT :"Line";O;
"not found!" :: stop !@P-
4 H=INT(E/G):: I=E-(G*H):: H
=H+G :: C=INT(L/G):: D=L-C*G
:: FOR E=E+4 TO F STEP 4 ::
CALL PEEK(E,A,B):: IF A=C A
ND B=D THEN 6 !@P-
5 NEXT E :: PRINT :"Line";N;
"not found!" :: STOP !@P-
6 E=E+3 :: J=INT(E/G):: K=E-
(G*J):: J=J+G :: CALL LOAD(-
31952,H,I,J,K,):: STOP !@P-
```

from LEHIGH 99'ER COMPUTER GROUP JAN'85

A CALL LOAD FOR A LOAD

I have been interested in finding a RUNable XBASIC CALL LOAD that would dependably RUN "DSK1.LOAD". Finally, one has turned up in the darnedest, ahmen amazing place. Among an abundance of public domain files and downloads we received in an exchange with the SIOUXLAND 99'ers, was the music program "Amazing Grace". Authored by Cecil Crowder, it plays the tune, then does the LOAD. I don't know how it works, but it does. The screen colors are legitimate, the title screen is skipped. In short, this CALL LOAD is the one.

```
CALL INIT :: CALL LOAD(-31961,149)
```

That equals >8327, and used by console's BASIC interpreter.

HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	206	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

CONVERT BY THE LOOK
by Jack Schreiber
LEHIGH 99'ER COMPUTER GROUP JAN'85

As we learn more about programming the need to convert Hexadecimel to decimal because a necessaity. The conversion shouldn't be a stumbling block -- we can use a conversion table. Our particular table will convert any decimal number from 0 to 65535. or four hex digits. The table is easy to use and is set up with four columns. Keep in mind that the Hex numbering system is a base 16 system as compared to our standard decimal base 10 system. How do we get 16 digits for a base 16 system? By using the letters A through F to represent values of 10 to 15; thus 12 is equal to C in hex. The higest digit in hex is F, the 16th didgit in a base 16 system. Look at column 2 we find decimal 16=10 Hex. Note we added the zero. (Hex is positional, just like ordinary decimal.)
For 17 decimal we use the 10 in Hex + 1 from colum 1 = 11 in Hex.
Lets try a conversion with four Hex digeits example:
Hexadecimal 1234= ? another: AFFF= ? 4096 - from column 4,40960 - from column 4,512 - from column 3,3840 - from column 2,240 -from column 1,15 - 45055 in decimal
Alternatively, we can use the table backwards. To convert Dec to Hex we locate our number directly. If our number falls between two numbers on the table, we take the lower of the two numbers. We then proceed to the next column, selecting the number or span that comes closest to the remainder.

TRANSL

VAR 80 to 163 conversion pgm

prog idea by John Hamilton
Central Iowa 99/4A USERS GROUP
modified by Fred Hawkins,
Lehigh 99'er Computer Group
from Lehigh 99'er Computer Group News Letter AUG84

Each line in the program must have a line number. ERROR trapping on lines 2 and 6 will discard any line number. The program text line numbers DO NOT have to be in order; the MERGE command will put them where they ought to go. If you use TI-WRITER, be sure to use the FIXED cursor mode. Otherwise each line will have a carriage return affixed to it and you'll get a syntax error. TI-WRITER will also save a TAB line if you use Word Warp mode. Keep lines shorter than 80 characters because TRANSL will clip off extras. (It is possable to get around that, so go ahead and change it.) You may modify line 1 to OPEN diffrent files. As the program RUN's each line is displayed to the screen. Watch for truncation on a LISTed or download file. When XBASIC reurns READY, type the following: NEW MERGE "DSK1.OUTR" ! or whatever If you RUN the MERGEd program now, nothing will happen. Each line is a trail REMark. Edit the program from the top down , using FUNCTN x and 1. As you delet each "!" and cursor-down, the XBASIC system retokenizes each line into a RUNable statment. If you get a syntax error, look first for a truncated line. Second, insure that you didn't use Word Wrap in TI-Writer. A quick check from XBASIC enviroment: CALL SCREEN(14)::FOR A=0 TO 12::CALL COLOR(A,16,15)::NEXT A::ACCEPT AT(4,4):A$

Don't answer the ACCEPT, but rather CLEAR it with a FCTN 4. Now LIST the program. Anything that looks like a red box is probably bad. TRANSL is a handy program. If you've a routine in a program, just LIST "DSK1.TESTR":(LINE NUMBERS), and RUN TRANSL aginst it. Voila! Here's your routine in a MERGE format. TRANSL can have all sorts of bells and whistles, like INPUT "text file?":a$, ect. Personally I prefer the straight- to - the-jugular approach for this utility.

```
1 CALL CLEAR :: OPEN #1:"DSK1.TESTR"
:: OPEN #2:"DSK1.OUTR",VARIABLE 163
2 LINPUT #1:L$ :: S=POS(L$," ",1)::
ON ERROR 6 :: N=VAL(SEG$(L$,1,S))
3 ON ERROR 5 :: A=INT(N/256)::
A$=CHR$(N-A6):: PRINT L$
4 PRINT #2:CHR$(A);A$;CHR$(131);
SEG$(L$,S+1,80); CHR$(0):: GOTO 2
5 PRINT #2:CHR$(255);CHR$(255)::
CLOSE #2 :: END
```

```
  CLOSE #2 :: END
  6 ON ERROR 5 :: RETURN 2
```

from LEHIGH 99'ER COMPUTER USER GROUP OCT'84
reprinted from the COLOMBUS, GA 994/A UG "REM"

```
1 ! TEXAS CURSOR
2 CALL CLEAR :: CALL INIT
3 CALL LOAD(8196,63,248)
4 CALL LOAD(16376,67,85,82,83,79,82,48,8)
5 CALL LOAD(12288,48,48,63,255,254,124,24,12)
6 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,8,4,32,32,
32,36,4,91)
7 CALL LINK("CURSOR")
8 INPUT X$ :: GOTO 8
```

RESequencing Index for Basic
LEHIGH 99'ER COMPUTER GROUP DEC'84

by Fredrick Hawkins

A program's size is a major handicap to the printer-less user. As the program gets bigger lines get harder to find. RESquencing is often dreaded as much as it's needed: 300 lines one can spend twenty minutes searching the LISTing just to fimd one routine. Well,here's a way to get elbow room by RESing, yet rapidly find the important lines:

```
10 GOTO 100
11 GOTO 230 REM first routine
12 GOTO 400 REM second
24 GOTO 534 REM third and so on
100 REM this is the first line
that executes
```

Line 10 lets your program RUN correctly. REM statments can be added in console BASIC because they tell the interpreter to ignore what follows; and mostly it does. The XBASIC version is simpler:

```
10 GOTO 100 ! auto index
11 GO 230 ! name or description
```

The GO is a variant syntax for GOTO. Its correct form GO TO, WHICH GETS tokenized as Hex 85 and B1 (85 & 177), GO by itself is an error but RES isn't bugged at all. GOTO has a single token, >86, which saves a byte. Console BASIC's editor won't let you ENTER just a GO but XBASIC is a litte more resiliant.

LEHIGH 99'ER COMPUTER GROUP DEC'84

XBASIC
BARE ESSENTIALS

by Fred Hawkins

NEKKID power
an EXBASIC utility program

y Fread Hawkins
(SAVE in MERGE format)

```
1 D=-31952 :: PRINT "CALL LO
AD(D,B,C)":"1st ";:: ACCEPT
A :: CALL PEEK(D+2,B,C):: C=
1+C-A*4 :: A=C<0 :: B=B+A ::
C=C-256*A :: STOP !@P-

2 D=-31950 :: PRINT "CALL LO
AD(D,B,C)":"last ";:: ACCEPT
A :: CALL PEEK(D-2,B,C) :: C
=C-1+A*4 :: A=C>256 :: B=B-A
:: C=C+256*A :: STOP !@P-

3 !@P-
```

Long time XBASIC users remember how welcome the text-file-to-MERGE-format (TRANSL, et al) program was when it first appeared. Like the shot heard'round the world, the conversion generated pograms, press, and abounding joy among TI people worldwide at near light speed.

Well, I've a comeback to TRANSL that will stop hearts, make faint men weak, litle old ladies giggle and strip out unwanted code, all from XBASIC's immediate mode.

Tha's right, while your program is standing there, you can get the beast and hang the parts out to dry, and leave the rest for dog food and garbage collection routines (XMLLNK >0036). The last is a red herring, so on with it. We'll start out with the whole pogram, ALL three lines and then decompose it a little, ending up with an in-the-head arithmetic problem. In all actuality, we've three programs here: line 1, line 2, and your program. To use Nekkid, you'll MERGE and either RUN 1 or RUN 2. If you RUN 1, you'll be able to RUN 2 but you can't do it the other way around -- there won't be a line 1 to RUN.

While we're mudding the waters, let's simplify some more: Two things prevent RUN 1, RUN 2 from executing your program. Firstly, you STOP ,which is handy for not RUNning 2 after 1. Second, you'll note that the prescan is turned off in all three lines. Line three makes certain that it is indeed off, as I don't quite trust its behavior with longish lines. Prescan off (!@P-) permits you to have define any of Nekkid variables as DIMs, DEFs or whatever and still RUN N ekid. There's a slim possiblity that some programs might turn the prescan back ON (!@+), but if your're smart enough to do that, you're smart enough to deal with it. Both of them, 1 and 2, work the same. Fifstly they remind you of what you'll have to ENTER in the immediate mode after they STOP. (That's the trick: really YOU do the work. Nekkid is an idiot savant -- great with numbers but does't DO anything.) Then they demand a number of lines that you want to save. RUN 1 will clip the end of your program, saving the '1st A ' lines, 2 will drop the start and save the 'last A' lines. Get it? An example (Diagram at bottom):

RUN 1 and get the reminders:
CALL LOAD(D,B,C)
1st ?

If you reply 6, Nekkid calculates the values of D, B, and C that will adjust the BASIC system's pointers for your program. When you ENTER 'CALL LOAD(D,B,C)',your program will suddenly consist of lines 1, 2, 3, 100, 110, and 120. For most ordinary purposes, the others are gone. RUN 2 is similar, except now you need to count backwards. A reply of 2 (after 1, above) will calculate the limits for 110 and 120. A simple diagram:

```
1 NEKKID 1 )
2   NEKKID 2      ) RUN  1
3   NEKKID 3 ) will save the last 1st A
lines
100...
110... your       )
120...program     )
140...is here.    )
150...chewing     )
```

```
160...its cud     )
170... and        )
180...minding     ) RUN 2
190...its own     ) will save the last A lines
200...business    )
```

Rember: RES 1,1 will make counting much easier.

How does it work? That's simple arithmetic and a little research. First of all we know that BASIC programs consist of a table of line numbers. These are made up of your program's numbers and their respective pointers to where the code of the line is stashed. Each line has 4 bytes, 2 for the number and 2 for the location. And secondly, we know that Hex 8330 and 8332 in PAD are pointers to the beginning and ending bytes of the table. >8330 is the pointer to the highest *line and* >8332 points to the lowest line. Changing them over to basic two's complement, we get -31952 and -31950.

So, without going into the absolute details, Nekkid looks at the value of the line number pointer, calculates how moany four byte locks are needed for 'A' lines and puts the calcuations into B and C. (That's RUN 1.) If there is a trick to all of this, it's that you can't just finish the program wilth CALL LOAD(D,B,C). Ordinarily -- or more accurately, every time I tried -- when you do adjust the line number table the system must forget where B and C are. In short, a fancy way to crash. But BASIC remembers in the immediate mode.

A simple immediate version: >CALL PEEK(-31952,A,B,C,D):: PRINT A,B,C,D >PRINT D-(**number of lines**)*4-1) If the calculation goes negative, add 256 and subtract one from C. CALL LOAD(-31952 with the two numbers. This is equivalent to Nekkid's RUN 1.

If you have a routine you need moved, just MERGE Nekkid, strip the routine out, RES it and SAVE it back in MERGE format. OLD your original and MERGE your new code. Or if you've a bunch to get rid of, strip off the front and SAVE that, then OLD to get the trail end. The middle disappears. MERGE the front back in.
Lastly, consider: Nekkid is fast! It puts TRANSL in the old age home, relegated to its proper role of TRANSLating download files. So let's get Nekkid!! (There goes a real hit single streaking by.)

Submitted by Michael Jones, CT99/4AUG

Who We Are:

We are a non profit organization whose membership is open to anyone interested in the activities of the group. All members pay annual dues of $15 . The membership year is one calendar year from receipt of your dues. You are invited to attend a couple of meetings before deciding if you wish to join.

OUR MEETINGS:

Our meetings are on the second Thursday of each month, at 7:30 PM, in the County Courthouse at the corner of 10th and San Antonio, on the second floor. Each meeting starts with club business and is followed by a demonstration or talk. The second meeting of the month is our special interest group, the Assembly Language SIG. Everyone is urged to share information on topics related to the TI 99/4A -- software review, hardware availability, programming tips, etc.

MEMBERSHIP:

You may join at any time of the year. Each membership "unit" has one vote in club matters and only one person from a "unit" may hold club office (on the other hand your entire family is invited to serve on club committees and participate at meetings and in meeting planning!). Our bi-monthly newsletter is mailed out the weekend before the meeting to serve as a reminder to attend the meeting. At meetings, you may buy 5 1/4 " disks for $7 each box. We buy these items in bulk quantity and charge cost plus enough to keep several in our library.

OUR LIBRARY:

Currently, our library is located at the librarian's house and will be available at each of our meetings. Our library has some 375 programs on disk and tape. Roughly half of the library's programs will run on just the console (some may require joysticks) and most of the others only require the Extended Basic module. We have programs of all types (games, education, scientific, and business and household management) and we poll the membership's interests before obtaining more.

Above all, we are a membership organization. We depend on everyone for directions the group should go and activities we should undertake. Join our group and share your ideas!

Newsletter Exchange

Our users group exchanges newsletters with several other recognized 99/4A Users Groups. The exchange is made with the understanding that, with proper credit to both the newsletter and author (if listed), your users group can reprint articles from our newsletter and, with proper credit, we can reprint articles from exchanged newsletters. (Please feel free to correct any typos, misspelling, bad grammar, etc.; we will try do the same.)

Please send your exchange newsletters to:

Central Texas 99/4A Users Group
P.O. Box 200246
Austin, Texas 78720-0246

Meetings Calendar

The following is a list of the currently scheduled meetings.

(April 9)(May 14)(June 11)(July 9)(August 13)(September 10) (October 8)(November 12)(December 10)

The meeting is scheduled to be held in the County Commissioner's Courtroom on the second floor of the Travis County Courthouse annex, which is located at 10th and San Antonio, in Austin Texas.
Meetings start at 7:30 PM and last until they throw us out!

The Adventures special interest group meets the Sunday following the meeting at Lori Milam's house. (4203 Yucatan, call for directions: 836-3301)

The Small C / Assembly language / SIG is held the Wednesday following the general meeting,at the Healthcare International building on Great Hills Trail off of 183 just north of 360.

Current Officers

President	Joe Pizzi	444-6829
	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
Vice President	Hank Kennedy	255-2812
Treasurer	Paul Dunn	258-4308
Secretary	Mike Schultz	835-2377
Librarian/		
Newsletter Editor	Mark Milam	836-3301

Correspondence for the group can be sent to:
the Central Texas 99/4A Users Group
Box 200246; Austin, Texas 78720.
or to the address given for the president.

Central Texas 99/4A Users Group
P.O. Box 200246
Austin, Texas 78720-0246

Edmonton 99'er Computer UG
P.O. Box 11983
Edmonton Alberta
Canada T5J 3L1